Chapter 3, Chapter 4

**Part 2. Textbook questions**

**Chapter 3.** [40 points]

3.1 Describe why an application developer might choose to run over TCP rather than UDP.

An application developer would choose TCP over UDP when they need a reliable data transfer. The inner workings of TCP use flow control, sequence numbers, acknowledgments, and times to ensure that data is delivered and intact! So if the information is critical and needs a guaranteed reliable transfer TCP is best. Where as a connection with UDP would not be guaranteed. Another concept would be congestion control, this would help an application developers project not eat up all the bandwidth.TCP allows for congestion control which regulates an applications bandwidth, essentially leaving wiggle room for other packets. UDP does not provide congestion control and quickly congest network bandwidth.

Reliable data transfer Elastic throughput Not time sensitive

3.2 Suppose host A is sending host B a large file over a TCP connection. If the acknowledge number for a segment of this connection is *y*, then the acknowledge number for the subsequent segment will necessarily be *y+ 1*. Is this true or false? Why?

False, all though it is true that the acknowledgment will be up one from the prior packets sequence number. The next sequence number will be comprised of a totally new number and its corresponding acknowledgement packet will be up one from that new sequence number as well. For example, packet#1 sends SEQ 100 and ACK 0 ,packet#2 returns a SEQ110 and ACK 101.Then packet number#3 could have SEQ125 and ACK111. This does not follow the y+1 rule because the next packets sequence number affects the next packets acknowledgment number. Not the previous packets ACK effecting the next packets ACK as the question implies.

Note: The answer is true only when 1) the data field of the segment in response to the first one has a length of 1; 2) the cumulative acknowledgement mechanism does not come into play; and 3) there are no retransmitted packets (either from A or from B) before the subsequent segment is sent. The example shown in the textbook is just a special case.

3.3 Suppose 5 TCP connections are present over some bottleneck link of rate X bps. All connections have a huge file to send (in the same direction over the bottleneck link). The transmissions of the files start at the same time. What is the transmission rate that TCP would like to give to each of the connections?

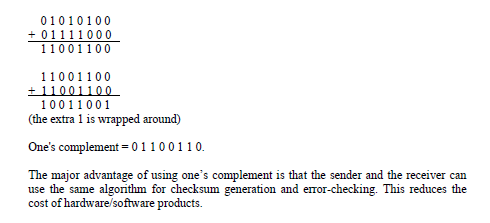
The formula is (Rate / Connections). So at a rate of (X bps / 5 connections) = X/5

3.4 How to identify a UDP socket? How to identify a TCP socket? Are these data fields same? Why?

UDP: dest IP address, dest port number TCP: source IP address; source port number; dest IP address; and dest port number TCP needs acknowledge and retransmission

3.5 UDP and TCP use 1's complement for their checksums. Suppose you have the following three 8-bit words: 10010101, 01111000, 10101011. What is the 1's complement of the sum of these words? Show all work. Why UDP takes the 1's complement of the sum, that is, why not just use the sum?

vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv



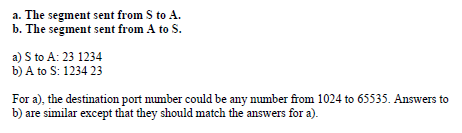
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

3.6 Suppose Client A initiates a HTTPS session with server S. Provide possible source and destination port numbers for:

a. The segment sent from S to A. =23 -1234

b. The segment sent from A to S. 1234 -23

vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv

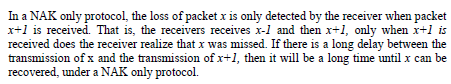


^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

3.7 Compare two pipelining protocols shown in the textbook – go-back-N and selective repeat.

3.8 In our textbook, protocol rdt 3.0 shows a data transfer protocols that uses only acknowledges. As an alternative, consider a reliable data transfer protocol that uses negative acknowledgements. Suppose the sender sends data only infrequently. Will a NAK-only protocol be preferable to protocol that uses ACKs? Why? Suppose the sender has a lot of data to send and the end-to-end connection experiences few losses. In the second case, would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?

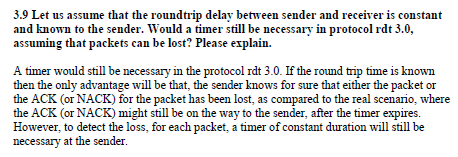
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv



^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

3.9 Let us assume that the roundtrip delay between sender and receiver is constant and known to the sender. Would a timer still be necessary in protocol rdt 3.0, assuming that packets can be lost? Please explain.

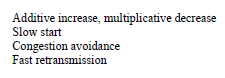
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv



^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

3.10 Briefly discuss the basic mechanisms adopted by TCP congestion control.

vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv

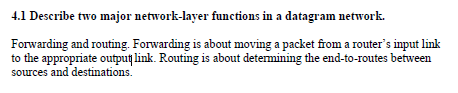


^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

**Chapter 4** [40 points]

4.1 Describe two major network-layer functions in a datagram network.

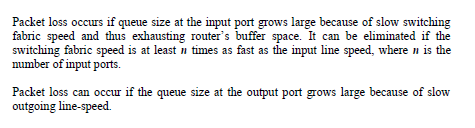
Vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv



^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

4.2 Describe how packet loss can occur at input and outputs of a router. Is it possible to eliminate packet loss at these ports? If so, how? If not, please explain.

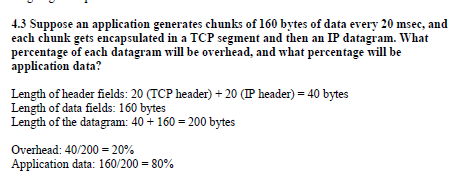
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv



^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

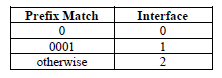
4.3 Suppose an application generates chunks of 1360 bytes of data every 20 msec, and each chunk gets encapsulated in a TCP segment and then an IP datagram. What percentage of each datagram will be overhead, and what percentage will be application data?

Vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv



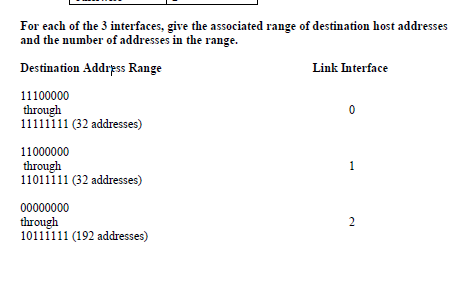
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

4.4 Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:



For each of the 3 interfaces, give the associated range of destination host addresses and the number of addresses in the range.

Vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv



^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

**3. Practical assignment** [20 points]

For this assignment, you can choose to use either Java or Python.

Please submit the following items in a ZIP file.

1) Source code;

2) Instructions on how to install and run your program;

3) A brief design document explaining your solution.

**Note**: I shall not provide remedial help concerning coding problems that you might have. Students are responsible for the setup of their own coding environment. Each student is also expected to debug their code. In addition most SMTP servers (e.g., NSU’s email server at nsusmtp.nova.edu) require authentication before sending messages. You can either hard-code the email account’s authentication information into the source code, or create a dummy or a free SMTP server (shown as follows) to test your program.

http://www.softstack.com/freesmtp.html

https://www.hmailserver.com/

<http://sourceforge.net/directory/os:windows/freshness:recently-updated/?q=smtp%20server>

(There are a couple of options. It seems that SMTPMail is a viable option if you feel comfortable with common line mode.)

**Sending Email with Java**

Java provides an API for interacting with the Internet mail system, which is called JavaMail. However, we will not be using this API, because it hides the details of SMTP and socket programming. Instead, you should write a simple Java program that establishes a TCP connection with a mail server through the socket interface, and sends an email message. You can place all of your code into the main method of a class called *EmailAgent*. Run your program with the following simple command:

java EmailAgent

This means you will include in your code the details of the particular email message you are trying to send.

Here is a skeleton of the code you'll need to write:

import java.io.\*;

import java.net.\*;

public class EmailAgent

{

public static void main(String[] args) throws Exception

{

// Establish a TCP connection with the mail server// Create a BufferedReader to read a line at a time.

InputStream is = socket.getInputStream();

InputStreamReader isr = new InputStreamReader(is);

BufferedReader br = new BufferedReader(isr);

// Read greeting from the server.

String response = br.readLine();

System.out.println(response);

if (!response.startsWith("220")) {

throw new Exception("220 reply not received from server.");

}

// Get a reference to the socket's output stream.

OutputStream os = socket.getOutputStream();

// Send HELO command and get server response.

String command = "HELO alice\r\n";

System.out.print(command);

os.write(command.getBytes("US-ASCII"));

response = br.readLine();

System.out.println(response);

if (!response.startsWith("250")) {

throw new Exception("250 reply not received from server.");

}

// Send MAIL FROM command.

// Send RCPT TO command.

// Send DATA command.

// Send message data.

// End with line with a single period.

// Send QUIT command.

}

}

For this assignment, you are required to use command-line-based Java and should not rely on any features provided by IDE such as NetBeans, Eclipse, etc. In your submission please send me a stand-alone document named “EmailAgent.java”. No executables should be submitted. No graphical interface should be used by your program.



Submitted to: Professor Wei Li

Student’s Name: Eric Webb

Date of Assignment: 09/29/2019

Title of Assignment: Assignment No. 2

Certification of Authorship: I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas, or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for this course.

Student's Signature: ERIC WEBB